

Available online at [www.sciencedirect.com](http://www.sciencedirect.com)**ScienceDirect**

Procedia CIRP 25 (2014) 400 – 405

[www.elsevier.com/locate/procedia](http://www.elsevier.com/locate/procedia)

8th International Conference on Digital Enterprise Technology - DET 2014 – “Disruptive Innovation in Manufacturing Engineering towards the 4th Industrial Revolution”

## Minimizing Energy Consumption for Robot Arm Movement

Abdullah Mohammed<sup>a\*</sup>, Bernard Schmidt<sup>b</sup>, Lihui Wang<sup>a</sup>, Liang Gao<sup>c</sup>

<sup>a</sup>*KTH Royal Institute of Technology, Brinellvägen 68, Stockholm 100 44, Sweden*

<sup>b</sup>*University of Skövde, Högskelevägen, Skövde 541 28, Sweden*

<sup>c</sup>*Huazhong University of Science and Technology, 1037 Luoyu Road, Wuhan 430074 Hubei, China*

\* Corresponding author. Tel.: +46-8-790 90 16. E-mail address: [abdullah.mohammed@itk.kth.se](mailto:abdullah.mohammed@itk.kth.se)

### Abstract

Robots are widely used in industry due to their efficiency and high performance. Many of them are operating in the manufacturing stage of the production line where the highest percentage of energy is consumed. Therefore, their energy consumption became a major focus for many robots manufacturers and academic research groups. Nevertheless, the optimization of that consumption is still a challenging task which requires a deep understanding of the robot's kinematic and dynamic behaviors. This paper proposes an approach to develop an optimization module using Matlab® to minimize the energy consumptions of the robot's movement. With the help of Denavit-Hartenberg notation, the approach starts first by solving the inverse kinematics of the robot to find a set of feasible joint configurations required to perform the task, solving the inverse kinematics is usually a challenging step which requires in-depth analyses of the robot. The module then solves the inverse dynamics of the robot to analyze the forces and torques applied on each joint and link in the robot. Furthermore, a calculation for the energy consumption is performed for each configuration. The final step of the process represents the optimization of the calculated configurations by choosing the one with the lowest power consumption and sends the results to the robot controller. Three case studies are used to evaluate the performance of the module. The experimental results demonstrate the developed module as a successful tool for energy efficient robot path planning. Further analyses for the results have been done by comparing them with the ones from commercial simulation software. The case studies show that the optimization of the location for the target path could reduce the energy consumption effectively.

© 2014 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license

(<http://creativecommons.org/licenses/by-nc-nd/3.0/>).

Peer-review under responsibility of The International Scientific Committee of the 8th International Conference on Digital Enterprise Technology - DET 2014 – “Disruptive Innovation in Manufacturing Engineering towards the 4th Industrial Revolution”

**Keywords:** energy consumption; path planning, robot dynamics.

### 1. Introduction

Many industrial countries witnessed an increase in prices of both electricity and fuel during the last decade. According to the recent statistics one of the large consumers of energy is manufacturing industry. The majority of the energy is usually consumed by robots used in the manufacturing industry. In addition, the optimal usage of energy in robots plays an important role in minimizing CO<sub>2</sub> emission in the production stage of a product's lifecycle.

Industrial robots are often seen as unsustainable equipment that demand a high level of energy consumption. On the other hand, those robots provide precision; strength and sensing

capabilities which can produce high quality end products [1]. Consequently the robotic energy consumption became a major objective for many research groups and robot manufacturers. Several researchers focused on defining tools to measure and analyze the robot's energy consumption. For example the work reported by [2] contributes to identifying energy efficient strategies in robotic applications. Others like [3] summarized different methods for energy-efficient use of common industrial robots. At the same time, many researchers presented robotic trajectory planning approaches that have the ability to optimize time and energy consumptions [4-7].

However, these approaches put a high priority on minimizing the movement time of a robot, which may not

necessarily provide a minimization of the energy consumption. The total energy consumed by the robot is usually affected by the required torque on each joint and inertia tensors of each link. Other researchers focused on optimizing a robotic manufacturing system as a whole [8, 9]. Despite the aforementioned efforts, minimizing robotic energy consumption remains a challenge and requires further investigations.

In this paper we introduce an approach to minimizing the energy consumption of an industrial robot's movements. It is achieved by developing a module that optimizes the robot's joint configuration to follow a certain trajectory defined by an operator. We evaluated the performance of the system by comparing the results of the optimization module with those of commercial simulation software of the robot. Three scenarios have been used to study the results of the developed module; with and without a payload at the robot's end-effector. Further experiments have been performed in different locations of the robot's workspace to identify the areas with the lowest level of energy consumption.

## 2. System overview

The developed system consists of 5 modules: the first one is responsible for preparing the desired trajectory of the robot with the desired velocity and acceleration. The second module is dedicated to solve the inverse kinematics of the trajectory. The third module calculates the forward and backward recursions to solve the inverse dynamics of the desired task. The energy consumption is calculated by the fourth module, introducing a set of robot's joints configurations to be optimized in the fifth module. Fig. 1 illustrates the developed modules which are implemented in MATLAB® software.

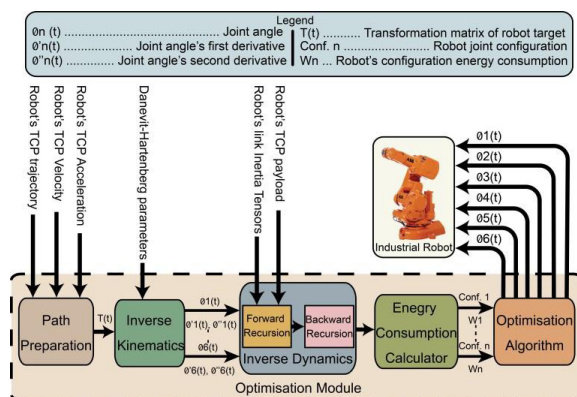


Fig. 1. Overview of the developed system.

## 3. Methodology

The implementation of the system starts by building the mathematical model of the robot to solve the kinematics and dynamics issues and calculate the robot's feasible joint configurations. The system then optimizes the robot's joints configurations to find the best one based on energy consumption.

Both the kinematic and dynamic models of the robot have been developed to analyze the forces and torques on the robot's joints during the motion of the robot. As an example the ABB IRB140 industrial robot has been chosen.

### 3.1. Denavit-Hartenberg notation

This development requires first establishing the kinematic parameters of the robot and assigning the joints frames. The definitions of the parameters are quite crucial for the kinematics and dynamics calculations in the forthcoming steps. The descriptions of the robot joint's frames were based on the Denavit-Hartenberg (D-H) notation. Starting from the base of the robot, the frames have been assigned for each joint. It is achieved by taking into account that the joint's rotation is always around the z-axis of the assigned frame. Fig. 2 shows application of the D-H notation on the robot model. Frame marked as ① is assigned to the robot base (link 0) and is used as global/world frame. Frames ④, ⑤, ⑥ create the robot wrist and their origin is located in the same place, called wrist centre  $P_w = [P_{xw} \ P_{yw} \ P_{zw}]^T$ . Parameters  $a_1$ - $a_4$ ,  $d_1$  and  $d_2$  are used to describe translation between frames in applied D-H notation.

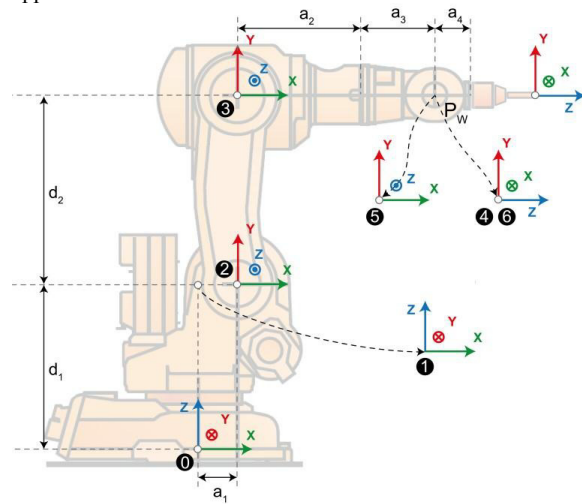


Fig. 2. Robot's frames assignments.

### 3.2. Forward kinematics

The forward kinematics is performed to calculate the transformation matrix that describes the location and orientation of the end-effector of the robot with respect to its base. With the help of the defined D-H parameters, the matrix is found by first calculating the transformation matrices of each joint with respect to the previous ones, and then multiplying eq. (1) those matrices. The result is a transformation matrix that describes the relationship between the end-effector and the base of the robot.

$${}^0T(\theta_1 \dots \theta_6) = {}^0T(\theta_1) \cdot {}^1T(\theta_2) \cdot {}^2T(\theta_3) \cdot {}^3T(\theta_4) \cdot {}^4T(\theta_5) \cdot {}^5T(\theta_6) \cdot {}^6T \quad (1)$$

$${}^0T(\theta_1 \dots \theta_6) = \begin{bmatrix} {}^{TCP}R & P \\ 0 & I \end{bmatrix}$$

Where:  ${}^i_jT$  is transformation matrix between link  $i$  and  $j$ ,  ${}^{TCP}{}^0T$  is transformation applied by tool attached to link6 of the robot.  ${}^{TCP}{}^0R$  and  $P$  are respectively rotation matrix and translation vector. In forward kinematics, transformations are functions of joint values.

### 3.3. Inverse kinematics

Solving the inverse kinematics of the robot is an important step for calculating the inverse dynamics of the robot in the forthcoming steps. It is archived under the assumption that the first three joints are responsible for the end-effector's position and the last three are responsible for its orientation. With given position and orientation of the target location, the position of the wrist is calculated. The pervious assumption provides a simplification for the calculations therefore it is widely used for solving the inverse kinematics of the serial-chain manipulators. The calculation of the first three joints ( $\theta_1, \theta_2, \theta_3$ ) is made

geometrically, whereas the last three joints ( $\theta_4, \theta_5, \theta_6$ ) are calculated analytically.

The process starts by solving the configuration of the first joint  $\theta_1$  in eq. (2). It is achieved by looking at the robot from the top view as seen in Fig. 3; which clearly shows that the angle changes the robot's wrist position in the x-y plane. Therefore,  $\theta_1$  is calculated using atan2 function defined by eq. (3) and the results is angle value within a range:  $[-\pi; \pi]$ . For joint 1 there are two possible values, that correspond to the configurations that robot arm is reaching the target toward or backward. Having more than one value for some of the robot's joints provides inverse kinematic solutions with multiple configurations.

$$\theta_1 = \begin{cases} \text{atan2}(P_{yw}, P_{xw}) & x > 0 \\ \text{atan2}(-P_{yw}, -P_{xw}) & x < 0 \end{cases} \quad (2)$$

$$\text{atan2}(y, x) = \begin{cases} \arctan(y/x) & x > 0 \\ \arctan(y/x) + \pi & y \geq 0, x < 0 \\ \arctan(y/x) - \pi & y < 0, x < 0 \\ +\pi/2 & y > 0, x = 0 \\ -\pi/2 & y < 0, x = 0 \\ \text{undefined} & y = 0, x = 0 \end{cases} \quad (3)$$

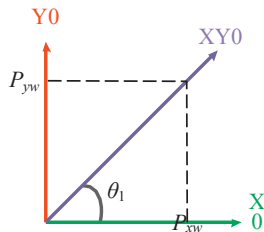


Fig. 3. The first joint angle projection on the x-y plane.

Then the values of the next two angles are calculated by looking at the plane Z0-XY0 formed by link2 and link3 of the robot in Fig. 4. It starts by calculating  $\cos\theta_3$  value using equation (4) which is used later to calculate joint 3 value  $\theta_3$  by implementing equation (5). Two values correspond to two robot arm configurations ①, ② as shown in Fig. 4.

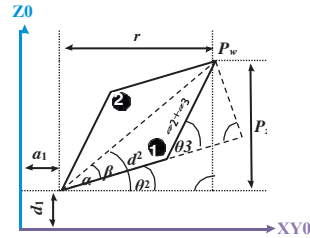


Fig. 4. The second joint and third joint angles projection.

$$\cos\theta_3 = - \frac{d_2^2 + (a_2 + a_3)^2 - (P_{zw} - d_1)^2 + r^2}{2(a_2 + a_3)d_2} \quad (4)$$

$$\theta_3 = \begin{cases} \text{atan2}\left(\sqrt{1 - \cos^2\theta_3}, \cos\theta_3\right) \\ \text{atan2}\left(-\sqrt{1 - \cos^2\theta_3}, \cos\theta_3\right) \end{cases} \quad (5)$$

The process then continues to calculate joint 2 value  $\theta_2$  according to equation (6) for each value of  $\theta_3$ .

$$\theta_2 = \beta - \alpha$$

$$\beta = \text{atan2}\left(\frac{P_{zw} - d_1}{r}\right)$$

$$\alpha = \text{atan2}\left(\frac{a_2 + a_3 \sin\theta_3}{d_2 + (a_2 + a_3) \cos\theta_3}\right) \quad (6)$$

When the first three joint values are obtained in all configurations, with forward kinematics the orientation of the wrist can be calculated (7) and the rotation that have to be applied in the wrist (8).

$${}^0R = {}^0R(\theta_1, \theta_2, \theta_3) \cdot {}^3R(\theta_4 = 0) \quad (7)$$

$$R_{zxz}(\theta_4, \theta_5, \theta_6) = {}^0R^T \cdot {}^{TCP}{}^0R \cdot {}^6R^T$$

$$= \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = R_Z(\alpha) R_X(\beta) R_Z(\gamma) \quad (8)$$

When Euler angles  $\alpha$ ,  $\beta$ ,  $\gamma$  in configuration ZYZ are used to express rotation  $\mathbf{R}_{ZYX}$ , the dependency between these angles and joint values is given in (9). To obtain Euler angles from known rotation matrix calculations follow (10-11).

$$\theta_4 = \alpha, \quad \theta_5 = -\beta, \quad \theta_6 = \gamma \quad \dots\dots\dots(9)$$

$$\beta = \begin{cases} \tan^{-1} \left( \frac{r_{31}^2 + r_{32}^2}{r_{33}} \right) \\ \tan^{-1} \left( -\frac{r_{31}^2 + r_{32}^2}{r_{33}} \right) \end{cases} \quad \dots\dots\dots(10)$$

$$\alpha = \tan^{-1} \left( \frac{r_{13}}{r_{23}} \right) \\ \gamma = \tan^{-1} \left( \frac{r_{11}}{r_{21}} \right) \quad \dots\dots\dots(11)$$

For each configuration of the first 3 joints, two possible configurations (10) in the wrist can be obtained. That gives in total eight possible configurations of joint values from inverse kinematics. Having at least one configuration means that the target is reachable by the robot.

### 3.4. Inverse dynamics

Due to its efficiency Recursive Newton-Euler Algorithm (RNEA) [10] is used to calculate the necessary parameters for the energy consumption calculation. With a given set of joint acceleration the algorithm is able to calculate the torque and force value of each joint. To start using the method, the inertial tensor matrixes values need to be obtained first. Therefore the 3D CAD model of the robot is used with the help of SolidWorks® software to calculate the inertial data. Calculations have been made with assumption that the mass of each robot link is distributed equally through the volume of the link.

Target path is interpolated with small time intervals. For each point on the path inverse kinematics is performed to obtain joint positions, velocities and accelerations for all possible configurations. After that, forward-backward recursion algorithm is applied.

#### 3.4.1. Forward recursion

This process is used to obtain the linear and angular motion of each link of the robot. It starts from the first link to the last one  $N=6$ , and with assumption that robot base (link0) is not moving, the initial conditions; angular velocity  $\omega_0$  and acceleration  $\alpha_0$  together with linear accelerations  $a_0$ ,  $a_{c0}$  are all set to be 0.

Then the angular velocity  $\omega_i$  and acceleration  $\alpha_i$  for each link  $i$  as well as linear accelerations at the end  $a_i$  and the gravity center  $a_{ci}$  of the link are also calculated using equations (12).

$$\omega_i = {}^{i-1}\mathbf{R}^T \cdot \omega_{i-1} + z_i \cdot \dot{\theta}_i \\ \alpha_i = {}^{i-1}\mathbf{R}^T \cdot \alpha_{i-1} + z_i \cdot \ddot{\theta}_i + \omega_i \times z_i \cdot \dot{\theta}_i \\ a_i = {}^{i-1}\mathbf{R}^T \cdot a_{i-1} + \dot{\omega}_i \times r_{i-1,i} + \omega_i \times (\omega_i \times r_{i-1,i}) \\ a_{ci} = {}^{i-1}\mathbf{R}^T \cdot a_{i-1} + \dot{\omega}_i \times r_{i-1,ci} + \omega_i \times (\omega_i \times r_{i-1,ci}) \quad \dots\dots\dots(12)$$

#### 3.4.2. Backward recursion

The forces and torques affecting the joints of the robot are calculated in this step. The process starts from the last link  $N$  and ends at the first link. The angular velocity and acceleration values which are obtained in the previous step are used in the current stage. First, the gravity vector  $g_0$  is expressed in frame for each link using equation (13).

$$g_i = {}^0\mathbf{R}^T \cdot g_0 \quad \dots\dots\dots(13)$$

Then the forces  $f_i$  and torques  $\tau_i$  are calculated for each link using equations (14). At this stage, inputs are external forces and torques  $f_{N+1}$ ,  $\tau_{N+1}$  applied to the robot end effector.

$$f_i = {}^i\mathbf{R} \cdot f_{i+1} + m_i (a_{ci} - g_i) \\ \tau_i = {}^i\mathbf{R} \cdot \tau_{i+1} - f_i \times r_{i-1,ci} + \\ + {}^i\mathbf{R} \cdot f_{i+1} \times r_{i,ci} + \omega_i \times (I_i \cdot \omega_i) + I_i \cdot \alpha_i \quad \dots\dots\dots(14)$$

### 3.5. Energy consumption calculating

The calculation of the power consumption is performed by two steps. First, the power consumed in each joint during a specific time interval  $k$  is found, and then the total power consumption at that interval is calculated by summing all joints power consumptions as shown in equation (15). For short time intervals, it can be assumed that torque and angular velocity are constant.

$$P(k)_i = (\tau_i(k) \cdot \dot{\theta}_i(k)) \\ P(k) = \sum_{i=1}^n P_i(k) \quad \dots\dots\dots(15)$$

The energy consumption is then calculated in equation (16), where  $d_{tk}$  is the length of  $k$ th time interval of the path divided into  $M$  intervals.

$$E = \int_{t_0}^{t_M} P(t) dt \cong \sum_{k=0}^M P(k) \cdot dt \quad \dots\dots\dots(16)$$

### 3.6. Energy optimization

The developed system is designed to choose the joint configuration that requires the lowest energy consumption. Since the number of possible configurations is limited to a few solutions; an optimisation algorithm is implemented to analyse the consumption in each configuration in iterative way through the whole target path and chooses the best one.

#### 4. Case studies

The validity of the optimisation module was examined using three scenarios: straight line, square path and different locations. These scenarios are explained in detail in the following sections.

##### 4.1. Straight line scenario

In this case, a straight line path beginning at (596, 0, 662), and ending at (596, -211, 580) mm was used as a desired trajectory for the end-effector of the robot.

Table 1. Straight line scenario results from the optimisation module.

| Start conf. | Start joint values | End conf.  | End joint values | Energy consumption [J] |             | Payload |
|-------------|--------------------|------------|------------------|------------------------|-------------|---------|
|             |                    |            |                  | Module                 | RobotStudio |         |
|             | 0                  |            | -21.29           |                        |             |         |
|             | -13.38             |            | -22.27           | 19.6                   | 28.7        | 0kg     |
|             | 7.42               |            | 6.28             |                        |             |         |
| (0,0,0,x)   | 0                  | (-1,0,0,x) | -55.28           |                        |             |         |
|             | 5.95               |            | 56.72            | 27.5                   | 38.4        | 6kg     |
|             | 0                  |            | 52.2             |                        |             |         |

Two experiments have been done to evaluate the performance of the module, with and without payload; both in the robot simulation software (RobotStudio®) and in our optimization module. Table 1 shows the results from the optimization module compared with the RobotStudio® ones.

##### 4.2. Square path scenario

A square shaped path is used as another case to evaluate the developed system. The system is again tested without and with payload. Optimal trajectory for the case with payload is presented on Fig. 5, where joint values are obtained through inverse kinematics, joint velocity and acceleration are calculated by differentiation and torques are obtained with inverse dynamics calculations.

Table 2 and Table 3 show the results from the optimization module and the ones from RobotStudio®, respectively. The differences in energy consumption calculated by our implemented module and RobotStudio® are caused by the simplification of used dynamic model.

Table 2. Square path scenario results from the optimisation module.

| Corners' joint values |        |        |        | Energy consumption [J] |           |
|-----------------------|--------|--------|--------|------------------------|-----------|
| 1                     | 2      | 3      | 4      | without load           | with load |
| -21.69                | -26.10 | -35.82 | -30.37 |                        |           |
| -37.83                | -27.45 | -33.22 | -43.19 | 68.7                   | 71.3      |

|        |        |        |        |                                       |
|--------|--------|--------|--------|---------------------------------------|
| -4.49  | -22.54 | -12.63 | 5.17   | Optimal configuration<br>(-1, 1, 0,x) |
| -30.56 | -32.60 | -45.16 | -43.57 |                                       |
| 46.61  | 54.73  | 55.62  | 47.18  |                                       |
| 22.08  | 20.27  | 29.59  | 32.88  |                                       |

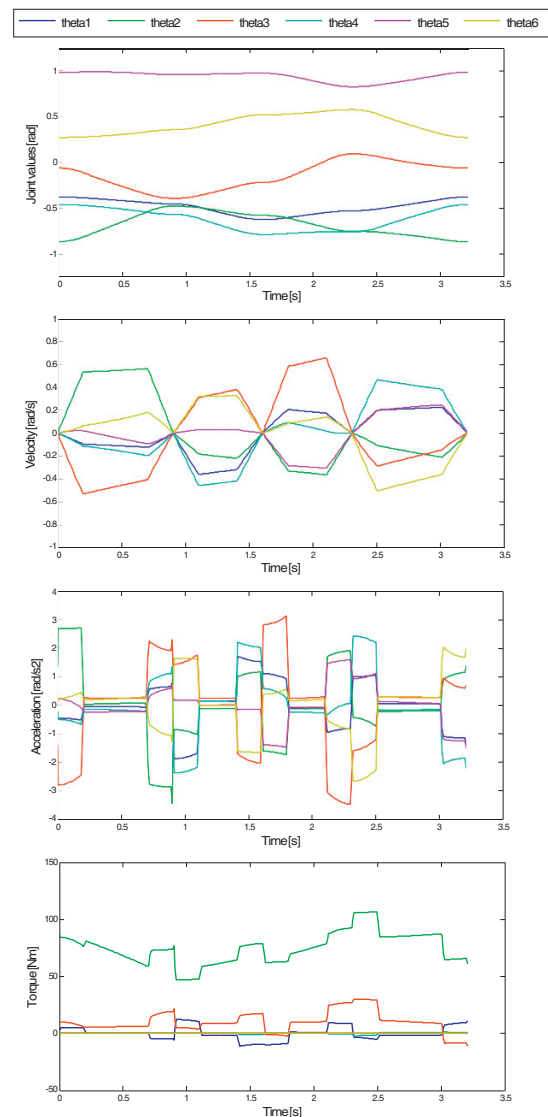


Fig. 5. Results obtained from the second case.

Table 3. Square path scenario results from RobotStudio®.

| Joint configuration | Energy consumption [J] |           |
|---------------------|------------------------|-----------|
|                     | without load           | with load |
| (-1,1,0,x)          | 80.5                   | 88.8      |
| (1,1,0,x)           | 150.0                  | 158.0     |
| (-1,-1,1,x)         | 104.9                  | 112.5     |



#### 4.3. Different locations scenario

In this scenario square horizontal path is placed in different locations in robot workspace. For each location the minimal energy consumption for optimal configuration is calculated. Fig. 6 presents how the energy consumption depends on the location of the target path in the robotic space. Data are visualized for slices  $x=0$ ,  $y=0$  and  $z=0$ . The color in the figure indicates the level of the power consumption. White points represent the locations where the task cannot be performed. It could be noticed that energy needed to perform the task varies significantly and is strictly related to the location of the target path. This scenario provides a tool for the robot cell designer to place the workpiece in the suitable location. The red points in the figure give indication to the designer to avoid the areas that consumes the highest level of energy.

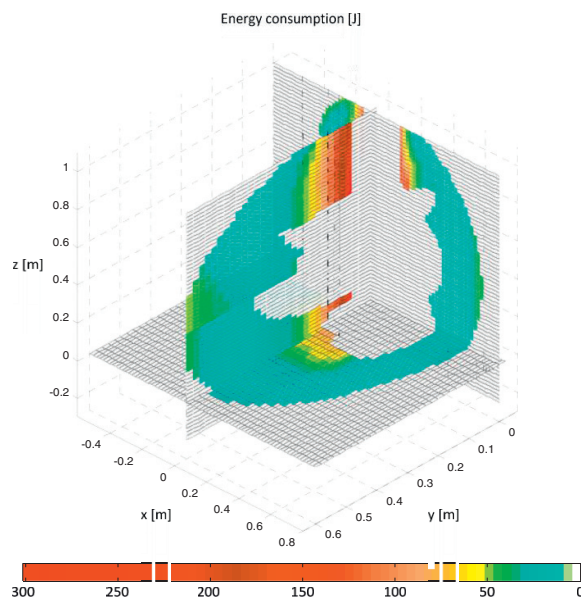


Fig. 6. Energy consumption analyses for robot workspace.

#### 5. Conclusions

It is important to optimize the energy consumption for the economic aspect as well as to minimize the potential environmental impact. The same target can be reached by the robot arm with different configurations and each configuration needs different power consumption. Our module addresses this issue by selecting the optimal configuration to perform the task.

As presented in the last case, the energy consumption for performing the task depends also on the location where the target path is defined. In this case the optimization of the location of the path to minimize the energy consumption seems to be required to ensure energy saving. The optimal location depends on the robot's dynamic model which is affected by the technical specifications and the task requirements.

Our future work will focus on the dynamic model of the robot, as it needs to be expanded to consider friction and the dynamics of the actuators that include motors, gears and couplings. The current implementation has a modular structure, and it can be easily updated with more accurate model as well as extended for other types of robots. Moreover, validation of energy consumption model with actual energy consumption through real experiment and measurements need to be performed. Furthermore, the developed energy optimization module can be extended in the future to consider the orientation of the target path.

#### Acknowledgments

This work is supported by GREENet project. The authors of this paper would like to thank Andrea Arzamendi Moñux and Miren Marticorena Lizaso for their partial contributions.

#### References

- [1] Bugmann, G., Siegel, M., Burcin, R., A Role for Robotics in Sustainable Development, In: Proceedings of the IEEE Africon, 2011, p. 13-15.
- [2] Chemnitz, M., Schreck, G., Kruger, J., Analyzing energy consumption of industrial robots, In: Proceedings of the IEEE ETFA'2011, 2011.
- [3] Meike, D., Ribickis, L., Energy Efficient Use of Robotics in the Automobile Industry, In: Proceedings of the 15th International Conference on Advanced Robotics, 2011, p. 507-511.
- [4] Hansen, C., Oltjen, J., Meike, D., Ortmaier, T., Enhanced Approach for Energy-Efficient Trajectory Generation of Industrial Robots, In: Proceedings of 8th IEEE International Conference on Automation Science and Engineering, 2012.
- [5] Meike, D., Ribickis, L., Industrial Robot Path Optimization Approach with Asynchronous FLY-BY in Joint Space, 2011 IEEE International Symposium on Industrial Electronics (ISIE), 2011, p.911-915.
- [6] Wei, Y., Minxiu, K., Lining, S., Optimal motion generation for heavy duty industrial robots-control scheme and algorithm, In: Proceedings of the 2011 IEEE International Conference on Mechatronics, 2011, p. 528-533.
- [7] Verschuer, D., Demeulenaere, B., Swevers, J., De Schutter, J., Diehl, M., Time-Energy Optimal Path Tracking for Robots: a Numerically Efficient Optimization Approach, AMC '08. 10th IEEE International Workshop on Advanced Motion Control, 2008, p. 727-732.
- [8] Vergnano, A., Thorstensson, C., Lennartson, B., Falkman, P., Pellicciari, M., Leali, F., Biller, S., Modeling and Optimization of Energy Consumption in Cooperative Multi-Robot Systems, IEEE Transactions on Automation Science and Engineering, 9:2, 2012.
- [9] Meike, D., Ribickis, L., Recuperated Energy Savings Potential and Approaches in Industrial Robotics, In: Proceedings of IEEE International Conference on Automation Science and Engineering, 2011, p. 299-303.
- [10] Luh, J.Y.S., Walker, M.W., Paul, R.P.C., On-line Computational Scheme for Mechanical Manipulators, Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME, 102:2, 1980, p. 69-76.